The Report committee for Richard Hsueh-Yee Leu

Certifies that this is the approved version of the following report:

Bayesian Analysis of the Complex Bingham Distribution

APPROVED BY

SUPERVISING COMMITTEE:

Paul Damien, Supervisor

Jennifer Duthie

Bayesian Analysis of the Complex Bingham Distribution

by

Richard Hsueh-Yee Leu, B.S.As.E, B.S.Phy.

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN STATISTICS

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2010

Dedicated to my wife Ivonne.

Acknowledgments

First and foremost, I owe my deepest gratitude to my report supervisor, professor Paul Damien. Without his guidance, support, and encouragement, this report would not be possible. I would like to thank Dr. Jen Duthie, the reader of the report, for her patience. I would like to also thank Ben Garcia, graduate advisor in the department of mathematics, for his help. I am grateful to my dissertation supervisor, professor Duane Dicus, for his support and flexibility. Lastly, I would like to acknowledge all my family and friends who encouraged me during the writing process.

Bayesian Analysis of the Complex Bingham Distribution

Richard Hsueh-Yee Leu, M.S. Stat.
The University of Texas at Austin, 2010

Supervisor: Paul Damien

While most statistical applications involve real numbers, some demand complex numbers. Statistical shape analysis is one such area. The complex Bingham distribution is utilized in the shape analysis of landmark data in two-dimensions. Previous analysis of data arising from this distribution involved classical statistical techniques. In this report, a full Bayesian inference was carried out on the posterior distribution of the parameter matrix when data arise from a complex Bingham distribution. We utilized a Markov chain Monte Carlo algorithm to sample the posterior distribution of the parameters. A Metropolis-Hastings algorithm sampled the posterior conditional distribution of the eigenvalues while a successive conditional Monte Carlo sampler was used to sample the eigenvectors. The method was successfully verified on simulated data, using both flat and informative priors.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In statistics we mostly work with real valued variables. Whether these variables are Boolean, categorical, or continuous, their values lie in the real space. However, some applications use complex numbers. For example, [33] and [21] model wind profiles with the use of a complex valued model. Wind speed and direction are combined into a single complex variable. In many cases, modeling with complex numbers involves taking real valued data and assigning some parts as real and others imaginary. See [34] for an introductory explanation on the advantages of using complex valued models for real valued data. Signal processing is one example. The operations on two input signals are modeled with one signal being real and the other imaginary. Some formal studies on complex valued signals were done by [49][40][17][44]. For a thorough explanation, see [45]. Applications of complex signal processing include brain EEG data [3][37], MRI image analysis[10], X-ray tomography[11], and communications[48].

Shape analysis is another discipline which uses complex variables. In shape analysis, the raw data is a set of Cartesian coordinates. The data is then transformed in order to remove extraneous information. The transformed

data only contains information about the shape. The resultant space lies in the complex plane. These geometric methods in statistical shape analysis were developed in [26][4][23]. Some techniques in this area are the Procrustes analysis and tangent space inference[13][6]. Procrustes analysis is analogous to a least squares problem; mean shapes are determined by minimizing the distance between the data and a regression-type model. Tangent space inference assumes small variability in the data. Therefore, the space around the data is approximately Euclidean, and normal statistical procedures are performed. Alternatively, we could model the data with a specific distribution. These distributions have supports in the complex space and have certain symmetries necessary for shape analysis. Chapter 6 in [13] discusses some of these distributions.

In this paper, we focus on a specific distribution, the complex Bingham. This distribution was introduced by Kent [28] and shown to be suitable for shape analysis. It is applicable only to two dimensional data. [28] also analyzed the maximum likelihood estimators (MLE) and compared them with other shape analysis techniques. The goal of this paper was to perform a full Bayesian analysis of the complex Bingham. In particular, a Monte Carlo Markov chain was implemented to sample the posterior. Section 2 reviews previous analysis done on the complex Bingham. Section 3 analyzes the prior and posterior distributions. Section 4 outlines the Gibbs sampler and demonstrates the validity of the algorithm.

Chapter 2

Complex Bingham Distribution

Statistical shape analysis has been applied in biology[35][14], medical imaging [5], archeology [7][27], geology[4], and image analysis[2]. Kendall [25] defines shape as the remaining geometrical information after location, scale, and rotation effects are removed. Data can be in the form of landmarks. Landmarks are a set of defined points on a shape that each specimen has. The positions of the landmarks are based on anatomical interest, mathematical properties, or arbitrarily choices. Statistical distributions are used to model landmark variations. The complex Bingham [28], complex Watson [36], and the complex angular central Gaussian [29][28] are some examples. Our analysis focused on the complex Bingham. The complex Bingham was introduced by Kent [28]. The normalization of the distribution was further examined in [32]. [1] studied bootstrap calibrated empirical likelihoods from a complex Bingham for shape estimation. [47] derived a discriminant rule based on this distribution. [39] showed that the complex Bingham is a conjugate prior for the complex Watson. The complex Bingham has been used in a variety of applications. The original Kent paper [28] studied rat vertebrae. The distribution has also been applied to brain surface images [15][16] and chromatin fiber structures[50]. In this section, some properties of complex vectors and

matrices were used. We refer readers to Appendix A.2.

2.1 Shape Analysis

Landmark data in 2 dimensions can be represented by a set of complex numbers. Each complex number, $z_i = x_i + iy_i$, is associated with a landmark where (x_i, y_i) are the Cartesian coordinates. We call this space the "raw" landmark space. To remove the geometric information about location and scale, the landmark data is transformed into the pre-shape space (or pre-shape sphere). This space is denoted $\mathbb{C}S^{k-2}$ where k is the number of landmarks. The support of shape distributions is the pre-shape space.

To transform the raw data, z_{raw}, into the pre-shape space

1. multiple raw data with sub-Helmert matrix. $z_H = H_{sub} * z_{raw}$.
2. normalize. $z = z_H/||z_H||$

The first step removes translation effects while the second step removes scaling effects. If z_{raw} were shifted or magnified, z would be unchanged. Two things should be noted. First, the dimensions of z are smaller than the raw landmark data by one.

$$z = (z_1, z_2, \cdots, z_{k-1})^T \in \mathbb{C}S^{k-2}$$

If there are k landmarks, then z is a vector of k-1 dimension. Second, z is normalized such that

$$\sum_{i=1}^{k-1} |z_i|^2 = 1$$

The sub-Helmert matrix is a (k-1)x k matrix. It is the Helmert matrix modulo the first row. The jth row of the sub-Helmert matrix is defined as

$$(h_j, \cdots, h_j, -jh_h, 0, \cdots, 0)$$

where

$$h_j = -\sqrt{j(j+1)}.$$

The jth row consists of h_j repeated j times followed by $-jh_j$ and then zeros.

The pre-shape space does not take away rotation. Therefore, only distributions that are invariant under $z \to ze^{i\theta}$ would be appropriate for shape analysis. It is not sufficient just to have support on the pre-shape space. See [13] for a full discussion on statistical shape analysis.

2.2 Analysis of the Complex Bingham Distribution

The complex Bingham distribution has a probability density

$$f(z|A) = c(A)^{-1} e^{z^* Az}, \quad z \in \mathbb{C}S^{k-2} \tag{2.1}$$

where A is a $(k-1)$ x $(k-1)$ Hermitian matrix, the support is the pre-shape space, and * is the complex conjugate transpose. Note that the density is invariant under rotation making it suitable for shape analysis. Since A is Hermitian, its spectral decomposition is

$$A = \sum_{i}^{k-1} \lambda_i \gamma_i \gamma_i^*$$

where the λ_i's are the eigenvalues and γ_i's are the eigenvectors.

Kent[28] points out two important characteristics. One, the normalization constant $c(A)$ only depends on the eigenvalues of A. Therefore $c(A) = c(\Lambda)$ where Λ is the set of eigenvalues. The normalization for non-degenerate eigenvalues has an analytic form.

$$c(\Lambda) = 2\pi^{k-1} \sum_{j}^{k-1} a_j \exp\lambda_j, \quad a_j^{-1} = \prod_{i \neq j} (\lambda_j - \lambda_i)$$

Two, A and $A + \alpha\mathbf{1}$ define the same distribution where $\mathbf{1}$ is the identity matrix and $\alpha \in \mathbb{C}$. The only difference is a shift in the normalization: $c(\Lambda) \to e^{\alpha} c(\Lambda)$. This characteristic stems from the fact that $z^* z = 1$. This property allows us to shift the eigenvalues arbitrarily. For the rest of this paper, we set the eigenvalues of -A to be $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{k-1} = 0$ This positive definition of the eigenvalues was done in [30]. By setting $\lambda_{k-1} = 0$, we have removed one degree of freedom leaving us with $k - 2$ eigenvalue parameters.

2.2.1 Classical Analysis

In this section, we outline the classical analysis done in [28]. In some situations, it is convenient to rotate the pre-shape space data to principal axis.

$$w_j = \gamma_j^* z$$

We then write w_j in Kent's polar coordinates.

$$w_j = \sqrt{s_j} e^{i\theta_j}$$

Note that because we are working in the pre-shape space, $\sum s_j = 1$. We then define the vector $s = \{(s_1, \cdots, s_{k-2})^T | s_j \geq 0\}$ and $\sum_{j=1}^{k-2} s_j \leq 1$.

When equation 2.1 is transformed into these polar coordinates, the θ's follow a uniform(0,2π) distribution, and the s's follow

$$f(s|\lambda) = \frac{1}{2\pi^{k-1}} c(\Lambda)^{-1} \exp\left(\sum_{j=1}^{k-1} -\lambda_j s_j\right),\ s \in \mathcal{S}_{k-2} \tag{2.2}$$

where the support is the (k-2) unit simplex, $\mathcal{S}_q = \{s : \sum_i^q s_i \leq 1\}$. Equation 2.2 depends only on the eigenvalues.

We denote the likelihood as $\mathcal{L}_{\mathcal{S}}(A)$. If there are N data points then the likelihood is

$$\begin{aligned}
\mathcal{L}_{\mathcal{S}}(A) &= \prod_i^N f(z_i|A) \\
&= c(\Lambda)^{-N} \exp\left(\sum z_i^* A z_i\right) \\
&= c(\Lambda)^{-N} \exp\left(\sum \mathrm{tr}\left(A z_i z_i^*\right)\right) \\
&= c(\Lambda)^{-N} \mathrm{etr}\left(A \sum \left(z_i z_i^*\right)\right) \\
&= c(\Lambda)^{-N} \mathrm{etr}\left(A\mathcal{S}\right)
\end{aligned} \tag{2.3}$$

where $\mathcal{S} = \sum z_i z_i^*$. z_i is a vector so $z_i z_i^*$ is a Hermitian matrix. We term $\mathcal{S}$ the "data s-matrix" or just "s-matrix." The log-likelihood is

$$\ell = \mathrm{tr}\left(SA\right) - N \mathrm{log} c(\Lambda).$$

For the eigenvectors, the MLE's are the eigenvectors of the data s-matrix. The MLE's for the eigenvalues are the solution to a set of non-linear equations.

$$\frac{\partial(\mathrm{log} c(\Lambda))}{\partial \lambda_j} = l_j / N$$

where l_j's are the eigenvalues of the s-matrix. The solutions to these equations are normally not analytic. Therefore, numerical methods are needed to find the MLE's of the eigenvalues.

2.2.2 Generating Random Samples

In [30] three methods were presented to numerically generate samples from the complex Bingham. For generating random points, equation 2.2 is a more convenient form of the density. Generating s only depends on the eigenvalues. θ is generated from a uniform distribution. The three methods for sampling from a complex Bingham were

1. Truncation to simplex
2. Acceptance-rejection on simplex
3. Uniform on simplex and truncated gamma on $[0, 1]$.

For simplicity we used the first method for this paper. The algorithm involves three steps

1. Generate $k-1$, $U_j \sim$ uniform$[0,1]$ random variables
2. Transform to $S'_j = -(1/\lambda_j)\log(1 - U_j(1 - e^{-\lambda_j}))$
3. If $\sum S'_j < 1$ then keep. Otherwise reject and go back to step 1.

Once the s's are generated, then it remains to sample the θ's and to form the w's. To obtain z, $z = \Gamma w$ where Γ is the unitary matrix whose columns are the eigenvectors.

2.2.3 MAP Analysis

Previous Bayesian analysis on the complex Bingham involved maximum a posteriori (MAP) estimators. MAP estimators are the Bayesian analog to MLEs. The values of the parameters which maximize the posterior are the point estimators. Micheas and Dey [39] used MAP estimators for the complex Bingham . In this case, the parameter of the complex Watson is a vector in the pre-shape space. The posterior distribution of this parameter follows a complex Bingham. Since the exact form of the posterior is known, credible intervals can be estimated by directly sampling the Bingham. This case is different than the one we studied here. In our case, the data comes from a complex Bingham. The posterior is a distribution of a matrix, not a vector. The complex Bingham can not be its own conjugate prior. In our case, sampling from the posterior will require a Markov chain Monte Carlo.

MAP estimators are not optimal in Bayesian analysis. Unlike classical analysis, the parameters are treated as random variables with probability distributions of their own. A full Bayesian analysis would involve estimates on the shape of the posterior density. MAP's only provide an estimate of the peak location. On the other hand, a Markov chain Monte Carlo samples the posterior distribution. These samples are used to estimate the shape of the

distribution. This method results in a full Bayesian estimate of the parameters.

Chapter 3

Bayesian Analysis

For a full Bayesian analysis, we need to examine the posterior distribution of the Hermitian parameter matrix A. Since the normalization term only depends on the eigenvalues, it is convenient to spectral decompose A.

$$A = \Gamma \Lambda \Gamma^*$$

where Γ is a matrix with columns made from the eigenvectors and Λ is the diagonal matrix of the eigenvalues. See appendix A.2 for more discussion. Since A is Hermitian, the eigenvalues are real and Γ is unitary. In this paper we move back and forth between A and $\{\Gamma, \Lambda\}$ based on convenience. Inference was then made on the joint posterior

$$\Gamma, \Lambda | \mathcal{S}.$$

Since we utilized a Gibbs sampler, we examined the conditional posteriors.

$$\Lambda | \Gamma; \mathcal{S}$$

$$\Gamma | \Lambda; \mathcal{S}$$

The likelihood's relationships to the conditional posteriors were examined first. Priors were then added to the analysis.

3.1 The Likelihood in Bayesian Analysis

The joint posterior and the conditional posteriors are proportional to

$$\begin{aligned} f(\Gamma, \Lambda|\mathcal{S}) &\propto f(\mathcal{S}|\Gamma, \Lambda)\pi(\Gamma, \Lambda) \\ &= \mathcal{L}_{\mathcal{S}}(\Gamma, \Lambda)\pi(\Gamma, \Lambda) \end{aligned}$$

where $\pi(\Gamma, \Lambda)$ is the prior and $\mathcal{L}_{\mathcal{S}}(\Gamma, \Lambda)$ is the likelihood. First, we examined the conditional posterior of the eigenvalues. In this case, the normalization term of the likelihood was very important. Using equation 2.3 for the likelihood, the conditional posterior became

$$\begin{aligned} f(\Lambda|\Gamma, \mathcal{S}) &\propto \mathcal{L}_{\mathcal{S}}(\Gamma, \Lambda)\pi(\Gamma, \Lambda) \\ &= c(\Lambda)^{-N}\text{etr}(\mathcal{S}A)\pi(\Gamma, \Lambda) \\ &= c(\Lambda)^{-N}\text{etr}(\mathcal{S}\Gamma\Lambda\Gamma^*)\pi(\Gamma, \Lambda). \end{aligned} \tag{3.1}$$

Regardless of the prior, no simple or recognizable form of the conditional posterior was seen. For this reason we decided on a Metropolis-Hastings algorithm to sample this density.

The conditional posterior of the eigenvectors is not dependent on the likelihood normalization term.

$$\begin{aligned} f(\Gamma|\Lambda, \mathcal{S}) &\propto \mathcal{L}_{\mathcal{S}}(\Gamma, \Lambda)\pi(\Gamma, \Lambda)) \\ &\propto \text{etr}(\mathcal{S}\Gamma\Lambda\Gamma^*)\pi(\Gamma, \Lambda) \\ &= \text{etr}(\Lambda\Gamma^*\mathcal{S}\Gamma)\pi(\Gamma, \Lambda) \end{aligned} \tag{3.2}$$

We ignore the prior at this point of the analysis. If we were in the real space, this would be a particular case of the Bingham- von Mises - Fisher (BvMF) distribution [31]. In the BvMF, the density has the same form as equation 3.2 except Γ and $\mathcal{S}$ would be their real analogs: orthogonal and symmetric matrices, respectively (see Appendix A.2). We denote the density in equation 3.2 as the complex BvMF distribution, $U \sim \mathbb{C}\text{BvMF}(\Sigma, B)$, with probability density

$$f(U|\Sigma, B) \propto \text{etr}(\Sigma\, U^*\, B\, U)$$

In this distribution, Σ is a diagonal real matrix, B is a Hermitian matrix, and U is a unitary matrix. $\mathcal{S}$ is always Hermitian by definition.

3.2 Priors and Posteriors

First, N in equation 3.1 is equal to the trace of $\mathcal{S}$.

$$\begin{aligned}\text{tr}(\mathcal{S}) &= \sum_i^N \mathcal{S}_{ii} \\ &= \sum_i^N \left(\sum_j |z_{i,j}|^2 \right) \\ &= \sum_i^N (1) = N\end{aligned}$$

For a flat prior, the joint posterior can be written as

$$f(A|\mathcal{S}) \propto c(\Lambda)^{-\text{tr}(\mathcal{S})} \text{etr}\,(\mathcal{S}A)$$

A natural choice for the prior is

$$\pi(A|Q) \propto c\,(\Lambda)^{-M}\,\text{etr}(QA) \tag{3.3}$$

where Q is a Hermitian matrix with trace M. We restrict M to be a whole number. When this prior was used, equation 3.1 for the conditional posterior of the eigenvalues became

$$\begin{aligned} f(\Lambda|\Gamma, \mathcal{S}, Q) &\propto \mathcal{L}_{\mathcal{S}}(\Gamma, \Lambda)\pi(\Gamma, \Lambda) \\ &\propto c\,(\Lambda)^{-N-M}\,\mathrm{etr}((\mathcal{S}+Q)A). \end{aligned}$$

Equation 3.2 for the conditional posterior of the eigenvectors became

$$f(\Gamma|\Lambda, \mathcal{S}, Q) \propto \mathrm{etr}((\mathcal{S}+Q)\Gamma\Lambda\Gamma^*)$$

This prior is a conjugate prior to the complex Bingham. In other words

$$f(\Gamma, \Lambda|\mathcal{S}, Q) \propto \pi(\Gamma, \Lambda|\mathcal{S}+Q)$$

Technically, Q could be any Hermitian matrix of the same dimension as $\mathcal{S}$ as long as its trace was a whole number. Naturally, we could regard Q as prior data. M would be the number of prior data. The larger the number of prior data, the stronger the prior. This statement makes both intuitive sense and mathematical sense as the posterior is dependent on the trace of Q and $\mathcal{S}$.

Chapter 4

Markov Chain Monte Carlo

In this section, we discuss the Markov Chain Monte Carlo(MCMC) techniques for sampling the posterior. The Gibbs sampler of the eigenvalues and eigenvectors is

1. Sample from $\Lambda|\Gamma, \mathcal{S}$
2. Sample from $\Gamma|\Lambda, \mathcal{S}$.

As seen in section 3.1, the conditional posterior on the eigenvalues is complicated by the normalization term. A Metropolis-Hastings (M-H) algorithm was implemented to sample this conditional. The conditional posterior on the eigenvectors was also discussed in section 3.1. We used a "successive conditional Monte Carlo" sampler for the eigenvectors. Section 4.5 provides the details of this method.

We first discuss some basic diagnostics of MCMC. Then we examine ways to parameterize the eigenvectors in order to isolate the independent degrees of freedom. Afterwards, each component of the Gibbs sampler is tested, starting with the Metropolis-Hastings algorithm and then the successive conditional sampling. In both those cases, we assume flat priors. Finally, the

Gibbs sampling of the joint posterior for both flat and fixed priors is validated. Validation involves running the MCMC on randomly generated data from a known Hermitian matrix. This procedure is done for $(k-1)=2$ and $(k-1)=3$ dimensions.

4.1 Diagnostics of Chains

Diagnostics are needed whenever MCMC techniques are implemented. While MCMC is shown mathematically to converge, we can only take a finite number of samples. There are no mathematically rigorous methods that tell us when we have sampled enough or how fast MCMC converges. Diagnostics usually check three things: burn-in, mixing, and convergence.

Burn-in refers to the beginning chain elements. MCMC algorithms require an initial guess of the parameters. Depending on how far away this guess is from the peak of the distribution, it may take some time before the chain is truly sampling. These initial steps are disregarded. The length of the burn-in depends on the initial guess and on the type of MCMC algorithm.

Figure 4.1 demonstrates burn-in behavior. The beginning parts of both chains do not explore the proper regions because the initial guess was not close enough. Therefore, the MCMC algorithm required a few steps to get there. Diagnostics of burn-in usually reduce to qualitative checks like this. We examined trace plots to check for burn-in effects in our analysis.

Mixing is how well the chain is exploring the parameter space. Mixing is

particularly important when M-H algorithms are used. We do not want a chain that stays at one location for a long time nor a chain that continually goes very far from the center of the distribution. Poor mixing can occur for two reasons. First, an algorithm might not allow transitions to all states in finite steps. In other words, the probability to go from one point in the parameter space to another is zero. This situation is disastrous. In this case the algorithm will not work. The second situation applies to M-H algorithms. In this case, proposal distributions are arbitrarily defined by the user to sample the parameter space (see section 4.4). Poor choices of the proposal distribution can lead to poor mixing. For diagnostics on mixing, trace plots can also be used.

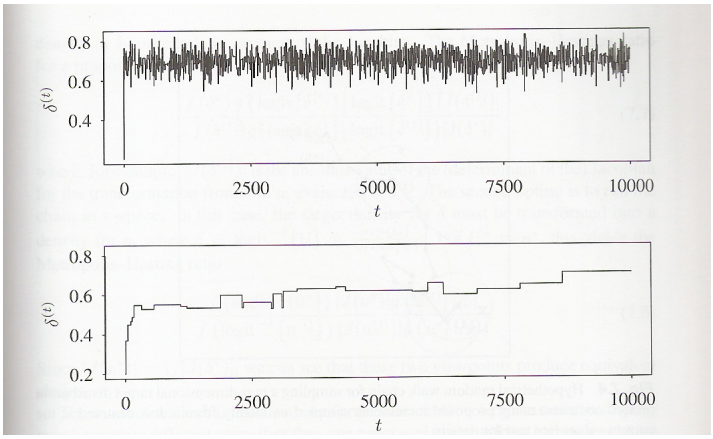

Figure 4.1: Examples of Good Mixing and Poor Mixing [20]

The bottom panel of figure 4.1 is an example of poor mixing. Multiple chain elements have the same value. This sticky behavior is undesirable. It indicates slow exploration of the parameter space. The upper panel shows good mixing. While these trace plots originate from random walks, the upper panel

seems to have no visual evidence of this. It appears to be random sampling. Note that at the smallest scales, MCMC's will appear as the bottom panel in figure 4.1. The Markovian behavior can not be eliminated on small scales. The goal is for the chain to be randomly sampling at large scales. This behavior is what is meant by good mixing.

Much of the discussion on MCMC practices focuses on convergence diagnostics. A wide range of convergence tests exist. For an overview see [12]. Some methods are based on quantitative analysis; others are visual. Some use multiple chains while others use a single chain. The theoretical justifications come from a variety of techniques. In our analysis, we checked convergence quantitatively with the Geweke test.

The Geweke convergence test[19] is based on statistical frequency domain analysis. The chain elements are modeled as a time series. This test assumes that a frequency density exists with no discontinuities at zero. The mean of the first part of the chain is compared to the mean of the latter part. The difference of the means converges to a normal distribution. The practical steps for the Geweke test are

1. Take 2 non-overlapping parts of the chain.
2. Calculate the means of these 2 sections.
3. Test if the means are from the same distribution. The null hypothesis is that the means do come from the same distribution.

4. The output is a z-score.

We say that the chain has converged if all the parameters have a z-score less than 2.

4.2 Parameterization of Unitary Matrix

For validation of our code, it was useful to parameterize the set of eigenvectors in such a way that can easily be analyzed. As we saw before, the eigenvectors form a unitary matrix. A NxN unitary matrix has N^2 degrees of freedom. However, eigenvectors are only known up to a scalar rotation. In other words $\gamma_i \equiv \gamma_i e^{i\phi}$ where ϕ is a real number. We fixed this degree of freedom by rotating each eigenvector such that its first component is positive real. This "gauging" reduces the degrees of freedom to $N^2 - N = N(N-1)$. One naive method is to just pick $N(N-1)$ components of the matrix, but these components would not be independent. Therefore, this method would not be ideal.

For the first test case, $(k-1) = 2$, there are two eigenvectors which form a 2x2 unitary matrix. We therefore have 2 degrees of freedom to examine. In this case, we parameterized the matrix to pick out the independent freedoms.

$$\Gamma = [\gamma_1, \gamma_2] = \begin{bmatrix} \sqrt{1-r^2} & r \\ r\, e^{i\theta} & \sqrt{1-r^2}\, e^{i(\theta \pm \pi)} \end{bmatrix} \tag{4.1}$$

This parameterization assures that the eigenvectors are orthogonal, unit normalized, and the first component real positive. The 2 degrees of freedoms are r and θ.

The second test case, $(k-1)=3$, was a little more complicated. We have 6 degrees of freedom. The unitary matrix of eigenvectors was parameterized as

$$\Gamma = [\gamma_1, \gamma_2, \gamma_3] = \begin{bmatrix} \sqrt{1-s^2-r^2} & \sqrt{1-a^2-b^2} & X \\ r\, e^{i\alpha} & b\, e^{i\gamma} & Y \\ s\, e^{i\beta} & c\, e^{i\delta} & Z \end{bmatrix} \tag{4.2}$$

where X,Y, and $Z \in \mathbb{C}$. We were not particularly interested in the parameterization of the 3rd eigenvector because it was completely determined by knowing the first two. 8 parameters are left but there are only have 6 degrees of freedom.

To reduce the number of parameters, we took the following steps. First, we found the null space basis of γ_1. See appendix B.2 for details. The null space of γ_1 is the set of vectors which are unit orthogonal to γ_1. The null space basis, N, is a 3x2 matrix. If we take a 2x1 unit vector and multiply it by N, the result is orthogonal to γ_1. Therefore, γ_2 and γ_3 are set to

$$[\gamma_2, \gamma_3] = Nz$$

where z is a 2x2 matrix with columns made up of unit vectors. To ensure that γ_2 and γ_3 are orthogonal to each other, we parameterized z similar to equation 4.1.

$$z = \begin{bmatrix} \sqrt{(1-a^2)} & a \\ a\, e^{i\gamma} & \sqrt{1-a^2}\, e^{i(\gamma \pm \pi)} \end{bmatrix} \tag{4.3}$$

The 6 degrees of freedom are: r, s, a, α, β, and γ.

The above procedure was adequate for the two test cases. However, as the dimensions increase, this procedure becomes complex. For example in

the $(k-1) = 4$ case, there are 12 degrees of freedom. The first eigenvector has 6 degrees of freedom. While it might be tempting to just use the first 2 eigenvectors to parameterize the 12 degrees of freedom, this choice is actually only 10 degrees of freedom. 2 degrees of freedom are lost since $\gamma_1 \cdot \gamma_2 = 0$. By following the $(k-1) = 3$ example, the construction of the null space basis of γ_1 results in z which is a 3x3 unitary matrix with 6 degrees of freedom. This is still too many. The first column of z has 4 degrees of freedom. Construction of the null space basis of this column results in z', a 2x2 unitary matrix. Parameterizing z' the same as equation 4.1 results in 2 degrees of freedom. This leads to the correct amount: $6 + 4 + 2 = 12$. We had to create 2 sets of null space basis. For a general NxN unitary matrix, we would have to calculate the null space basis $N - 2$ times in this cascading fashion.

4.3 Generating Test Cases & Validation of Algorithm

To test our algorithm, we generated 50 random points from a complex Bingham using the method outlined in section 2.2.2. The eigenvalues were set arbitrarily while the eigenvectors were randomly generated (see appendix B.3). These test cases acted as benchmarks to validate our code. For the first case, we have a 2x2 Hermitian matrix with 1 non-zero eigenvalue and 2 eigenvectors. Following the parameterization of the eigenvectors in equation 4.1, we have 3 degrees of freedom: λ, r, and θ. λ is set to 10. After generating our random 2x2 unitary matrix, $r = 0.51441$ and $\theta = 1.3261$. For the 50 randomly generated values, the MLE's are shown in the last column of table

4.1.

Parameter	Set Value	MLE
λ	10	10.2
r	0.514	0.527
θ	-1.33	-1.50

Table 4.1: $(k-1) = 2$ Parameters & Maximum Likelihood Estimates

For the second case, we have a 3x3 Hermitian matrix with 2 non-zero eigenvalues and 3 eigenvectors. The eigenvectors were parameterized as in equations 4.2 and 4.3. We have 8 degrees of freedom:λ_1, λ_2, r, s, a, α, β, and γ. The eigenvalues were set to $\Lambda = \{\lambda_1 = 100,\ \lambda_2 = 10\}$. The eigenvectors were found by randomly generating a unitary matrix. The resultant parameters can be seen in column 2 of table 4.2. The MLE's are listed in the last column.

Parameter	Set Value	MLE
λ_1	100	111
λ_2	10	9.4
r	0.930	0.935
s	0.330	0.312
a	0.791	0.764
α	0.821	0.863
β	3.14	3.17
γ	2.32	2.21

Table 4.2: $(k-1) = 3$ Parameters & Maximum Likelihood Estimates

4.3.1 Validation Procedure

To validate the MCMC algorithms, we compare the output with the computed likelihood surface. Most inferences on MCMC involve credible inter-

vals and kernel estimates of the marginal densities. Marginal densities become more complicated as the number of parameters increases. Also, we are more interested if the MCMC algorithm is exploring the shape of the posterior properly. Therefore, what we compared were the posterior vs. parameter plots. These plots were more indicative of whether the chains were truly exploring the posterior density.

MCMC's are interpreted as random samples from the posterior. If we plot the posterior density vs. a single parameter, we would expect the upper bound to be the maximum posterior. To illustrate, we sample 50 times from a normal distribution with mean $\mu = 100$ and standard deviation $\sigma = 20$. The resulting sample had a mean $\bar{x} = 98.75$ and standard deviation $s = 16.66$. Figure 4.2 shows the computed maximum likelihoods for the fixed values of the parameters. The left figure is for fixed values of μ, and the right figure is for fixed values of σ.

We expect the MCMC to sample values near this maximum and not venture too far from the peak of the likelihood. We also expect the MCMC output to have the same shape as figure 4.2. A simple M-H algorithm was used to sample μ and σ from the posterior $f(\mu, \sigma|\vec{x}) \propto f(\vec{x}|\mu, \sigma)$. Figure 4.3 shows the MCMC results. The shape of these plots are identical to figure 4.2. The algorithm also explored regions only around the likelihood peak. In this example, we are confident that the MCMC algorithm is sampling properly.

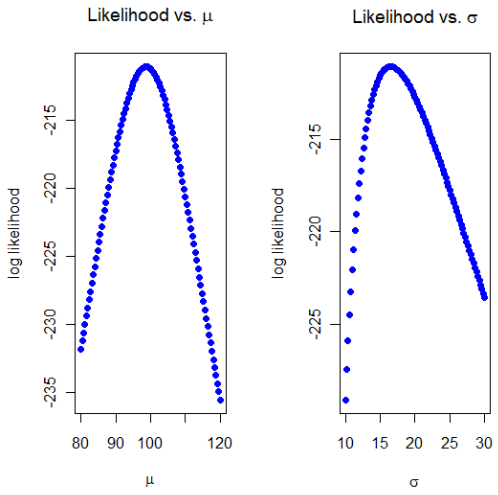

Figure 4.2: Calculated Maximum Likelihood For Fixed Parameter

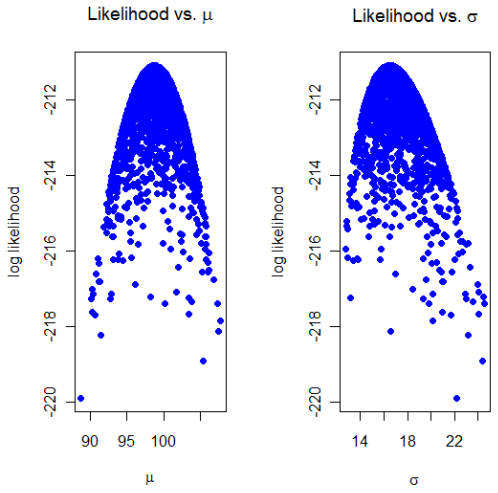

Figure 4.3: MCMC sampling of Likelihood

4.4 Metropolis-Hastings Sampling of Eigenvalues

Because of the complication of the conditional posterior of the eigenvalues, we used a M-H type MCMC for sampling. Other MCMC algorithms

would be difficult to implement since there seems no simple way to sample from this distribution. However, the likelihood and priors can be computed for a given set of parameters making M-H a suitable algorithm.

The general M-H algorithm for a set of parameters θ can be summarized in a few steps.

1. Draw a value in the parameter space from a proposal distribution $\rho(\theta|\theta^t)$ where θ^t is the previous chain value.

2. Calculate the posterior density at this point: $f(\theta|\text{data})$.

3. Calculate R: $R = \frac{f(\theta|\text{data})*\rho(\theta^t|\theta)}{f(\theta^t|\text{data})*\rho(\theta|\theta^t)}$

4. If $R > 1$, then θ is recorded into the chain. If $R < 1$, draw $u \sim \text{unif}(0, 1)$. If $u < R$, then record θ into the chain. Otherwise, copy θ^t as the next chain element.

The resulting chain can be considered a sample from $f(\theta|\text{data})$.

For a M-H sampler, the proposal distribution is arbitrary yet key to performance. As discussed before, the choice of the proposal effects how well the chain mixes. We set the proposal distribution as a truncated normal in a random-walk algorithm ($\rho(\theta|\theta^t) \rightarrow \theta^t + \rho(x|\text{mean} = 0, \text{sd} = \text{const})$). The mean of the normal distribution is the previous chain element. The variance is arbitrary and can be tuned for efficient "walking." The truncation was set to keep the λ_j's ordered and positive. With this kind of symmetric proposal,

R reduces to a ratio of the posteriors.

$$R \to \frac{f(\Lambda'|\Gamma, \mathcal{S})}{f(\Lambda^t|\Gamma, \mathcal{S})}$$

where Λ' are the proposed values of the eigenvalues. We no longer need to calculate the value of the proposal density. From testing, we found that a suitable standard deviation for the proposal was roughly one-tenth the initial guess of the eigenvalues. We used this value throughout.

For the $(k-1) = 2$ case, we fixed r, θ to the values in column 2 of table 4.1. Figure D.1 in the appendix shows the trace plots and density estimates of the chains. Based on the trace plots, the M-H algorithm showed good mixing and had no burn-in effect. The Geweke convergence test z-score was 0.97. Figure 4.4 shows a comparison between the computed likelihood curve and the resultant chain. The left panel is the computed log likelihood vs. λ. The right panel is the M-H output. The two plots are identical. We concluded that the M-H algorithm was sampling the conditional posterior of the eigenvalues.

For the $(k-1) = 3$ case, we fixed Γ to the values in column 2 of table 4.2. See figure D.2 in the appendix for trace plots. The Geweke z-scores for λ_1 and λ_2 were 1.86 and 1.33, respectively. Based on these plots, the M-H algorithm mixed well. Figure 4.5 shows a comparison between the computed likelihood contour and the resultant chain. The left two figures are the maximum log likelihood vs. λ_1 and λ_2. For a given value of λ_i, we calculate the maximum likelihood with the other eigenvalue being a free parameter. The right two figures are the M-H outputs. In this case, since we have more than

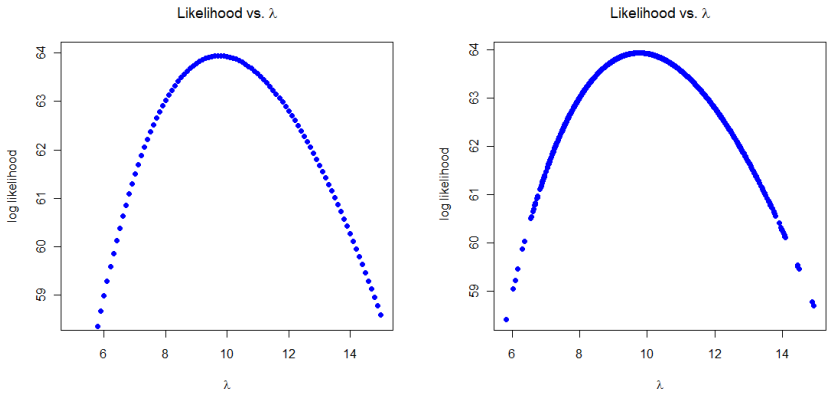

Figure 4.4: Computed Likelihood and M-H Output for λ

one eigenvalue we do not expect a smooth curve. The M-H algorithm should explore the values around the maximum likelihood. This includes points below the maximum likelihood curve. The shape and position of the curves are identical. We concluded that the M-H algorithm was sampling the eigenvalues properly for both cases. Table 4.3 compares the MCMC estimates of the

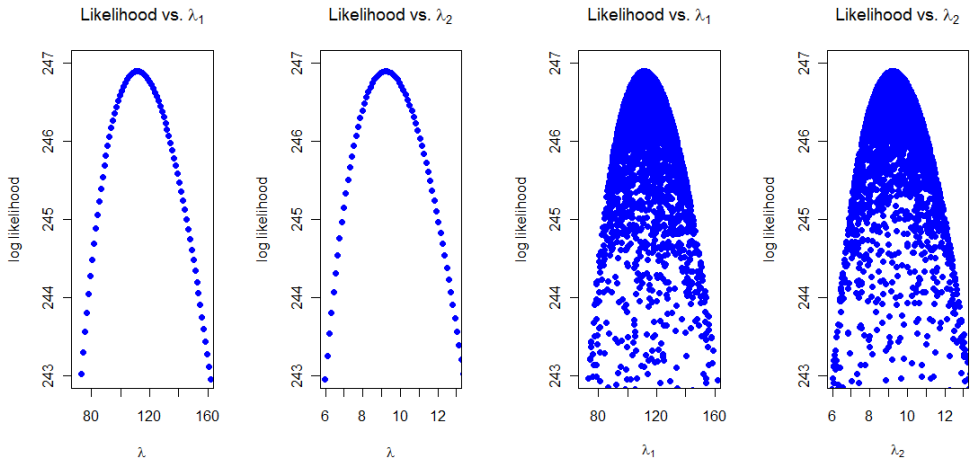

Figure 4.5: Computed Likelihood and M-H Output for Λ

eigenvalues with the MLE's. MLE's are not the same as in table 4.1 and 4.2. Here, we have the eigenvectors fixed where as in table 4.1 and 4.2 they were free parameters. MCMC estimates of the parameters were marginal estimates. The 2-σ intervals were based on the mean and standard deviation of the chains; we were assuming a non-skewed marginal density.

Parameter	M-H	MLE
$(k-1)=2$ Case		
λ	10.0 ± 2.8	9.8
$(k-1)=3$ Case		
λ_1	113 ± 31	112
λ_2	9.4 ± 2.6	9.2

Table 4.3: Parameter Estimates from M-H

4.5 Successive Conditional Monte Carlo Sampling of the Eigenvectors

In this subsection, we describe the sampling from the conditional posterior of the eigenvectors. From equation 3.2, the conditional posterior is

$$f(\Gamma|\Lambda,\mathcal{S}) \propto \operatorname{etr}\left(\Lambda\Gamma^{*}\mathcal{S}\Gamma\right)\pi\left(\Gamma,\Lambda\right)$$

At this point, we will assume a flat prior. For this section, it is convenient to examine equation 3.2 in terms of the individual eigenvectors.

$$f(\Gamma|\Lambda,\mathcal{S}) \propto \prod_{i} \exp\left(\lambda_i \gamma_i^{*}\mathcal{S}\gamma_i\right) \tag{4.4}$$

The joint eigenvector density can be written as a cascade of conditionals.

$$\begin{aligned}\Gamma|\Lambda,\mathcal{S} =&\gamma_1\cdots\gamma_{k-1}|\Lambda,\mathcal{S} \\ =&\gamma_{k-1}|\gamma_{k-2}\cdots\gamma_1,\Lambda,\mathcal{S} \\ &\times\gamma_{k-2}|\gamma_{k-3}\cdots\gamma_1,\Lambda,\mathcal{S} \\ &\vdots \\ &\times\gamma_2|\gamma_1,\Lambda,\mathcal{S} \\ &\times\gamma_1|\Lambda,\mathcal{S}\end{aligned} \tag{4.5}$$

Sampling the posterior becomes a matter of successively sampling each eigenvector based on the previous one. Eigenvector 2 is sampled conditioned on eigenvector 1. Eigenvector 3 is sampled conditioned on eigenvector 1 and 2 and etc. We term this method "successive conditional Monte Carlo" (SCMC) sampling.

Based on equations 4.4 and 4.5,

$$\gamma_1 \sim f(\gamma_1|\Lambda,\mathcal{S}) \propto \exp(\lambda_1\gamma_1^*\mathcal{S}\gamma_1)$$

This density function has the same form as the complex Bingham. Since γ_1 is unit normalized, it also has the same support. Therefore, sampling γ_1 reduces to sampling from $\mathbb{C}B(\lambda_1\mathcal{S})$. Continuing, based on equations 4.4 and 4.5,

$$\gamma_2 \sim f(\gamma_2|\gamma_1;\Lambda,\mathcal{S}) \propto \exp(\lambda_2\gamma_2^*\mathcal{S}\gamma_2) \tag{4.6}$$

such that $\gamma_1\cdot\gamma_2 = 0$. Note that γ_1 is not explicit in the density of γ_2 but in the support. Now the density of γ_2 is rewritten such that the density does take

into account the support. First, let N be the null space basis of γ_1. Then, $\gamma_2 = Nz$ and equation 4.6 becomes

$$\begin{aligned} f(\gamma_2|\gamma_1; \Lambda, \mathcal{S}) &\propto \exp(\lambda_2 \gamma_2^* \mathcal{S} \gamma_2) \\ &= \exp(\lambda_2 z^* N^* \mathcal{S} N z) \\ &= \exp(z^* (\lambda_2 N^* \mathcal{S} N) z) \\ &= \exp(z^* A_2 z) \end{aligned}$$

This density is a $\mathbb{C}B(A_2)$ with the orthogonality embedded inside A_2. Sampling γ_2 is then a matter of sampling $z \sim \mathbb{C}B(A_2)$ and then calculating $\gamma_2 = Nz$. We can continue this calculation for the rest of the eigenvectors.

1. Calculate the null space basis, N, for the previously sampled eigenvectors.
2. Sample z from $\mathbb{C}B(A_i)$ where $A_i = \lambda_i N^* \mathcal{S} N$
3. $\gamma_i = Nz$

Again, see appendix **??** for the computational methods of calculating the null space basis. These steps can be done for all the eigenvectors except the first eigenvector and the last eigenvector. As we have already seen, the first eigenvector is not conditional on any other eigenvector. We can forgo the calculation of the null space basis. The last eigenvector is conditional on all the other eigenvectors. This conditioning makes the last eigenvector completely determined; no sampling is necessary. The null space basis would be a $k - 1$ vector. All that is left is to rotate the first component so it is positive real.

We now present the results of our testing. Just as we did in testing the M-H, we show the trace plots and density estimates in the appendix. Also in the appendix are the Geweke convergence test scores. All chains showed good mixing, no burn-in effects, and convergence. For the two cases, we fixed the eigenvalues to the values in column 2 of table 4.1 and 4.2. For the $(k-1)=2$ case, the Unitary matrix is parameterized in the form of equation 4.1. For the $(k-1)=3$ case, the Unitary matrix is parameterize as in equation 4.2 and 4.3.

Figures 4.6 and 4.7 show the comparisons between the computed likelihood curves and the resultant chains. The left set of figures are the computed log likelihood vs. the parameters. The right set of figures are the MCMC output. The computed and chain plots were identical. We concluded that the successive conditional Monte Carlo algorithm was sampling the conditional posterior of the eigenvectors for both cases. Table 4.4 compares the MCMC estimates of the eigenvectors with the MLE's. Just as in the M-H eigenvalue sampling case, the MLE's are not the same as in tables 4.1 and 4.1. Here, we have the eigenvalues fixed to the set values. Again, the MCMC estimates of the parameters are marginal estimates. Intervals are at the 2-σ level.

4.6 Gibbs Sampling of Joint Posterior

The Gibbs sampler of the joint posterior is just the sequential sampling of the conditional posteriors: M-H sampling of the eigenvalues and SCMC sampling of the eigenvectors. We tested the Gibbs sampling for both flat

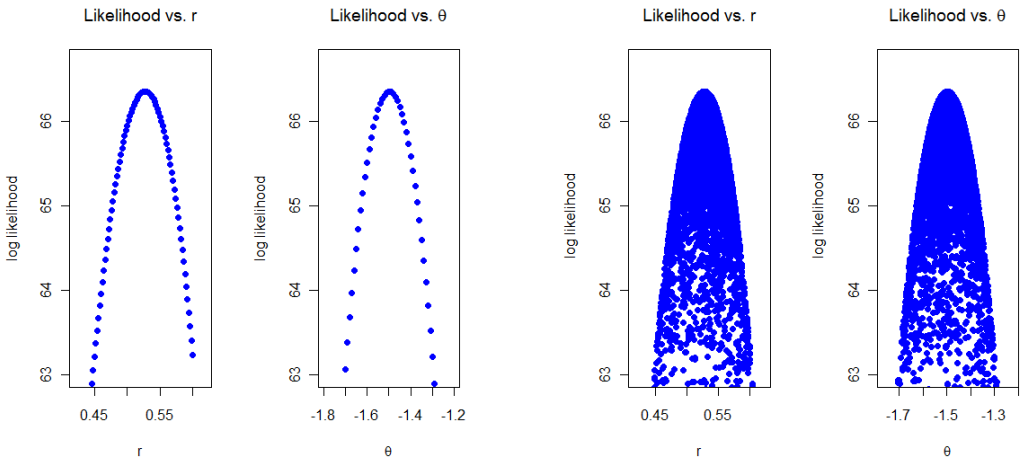

Figure 4.6: Computed Likelihood and SCMC Output of r and θ

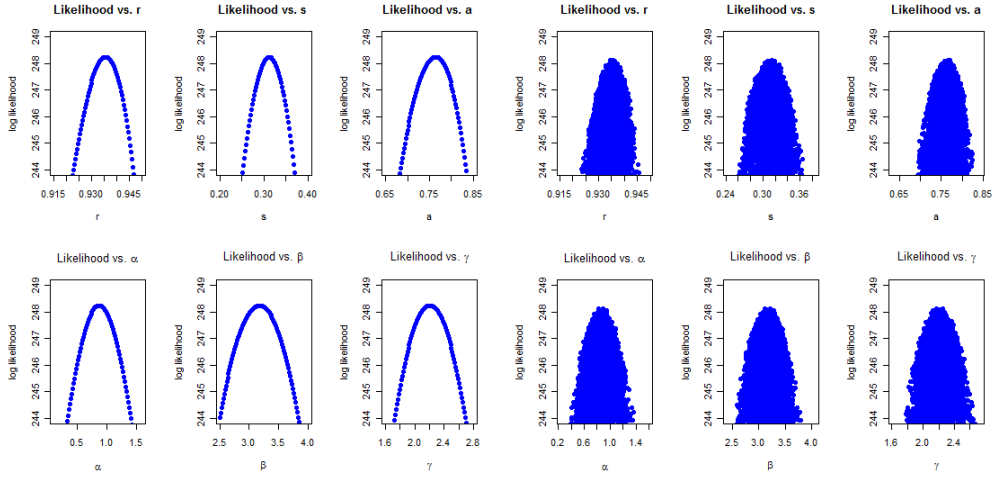

Figure 4.7: Computed Likelihood and SCMC Output for $(k-1) = 3$ Case

priors and informative priors. For flat priors the Gibbs sampling of the joint posterior is just an extension of the previous testing. For fixed priors we had to decide what the hyper-parameter should be.

Parameter	SCMC	MLE
$(k-1)=2$ Case		
r	0.528 ± 0.060	0.527
θ	-1.50 ± 0.16	-1.50
$(k-1)=3$ Case		
r	0.9344 ± 0.0079	0.9354
s	0.313 ± 0.038	0.312
a	0.759 ± 0.051	0.764
α	0.87 ± 0.33	0.87
β	3.18 ± 0.43	3.18
γ	2.21 ± 0.31	2.20

Table 4.4: SCMC Parameter Estimates of the Eigenvectors

We first analyze the flat priors case. The verification process is identical to the previous subsections. The trace plots and density estimates are in the appendix along with the diagnostics. Figures 4.8 and 4.9 compare the computed likelihood surfaces with the MCMC results for the $(k-1)=2$ and $(k-1)=3$ cases, respectively. Based on these plots, we concluded that the Gibbs algorithm was properly sampling the joint posterior. Table 4.5 shows the MCMC estimates of the parameters.

In section 3.2 Q was introduced as the hyper-parameter in the prior distribution. Q could represent prior data while $\mathcal{S}$ represent current data. This interpretation hints at a way to construct Q.

1. Set Λ_Q and Γ_Q. $A_Q = \Gamma_Q \Lambda_Q \Gamma_Q^*$
2. Sample N z's from $\mathbb{C}B(A_Q)$
3. Construct $Q = \sum_i^N z_i^* z_i$.

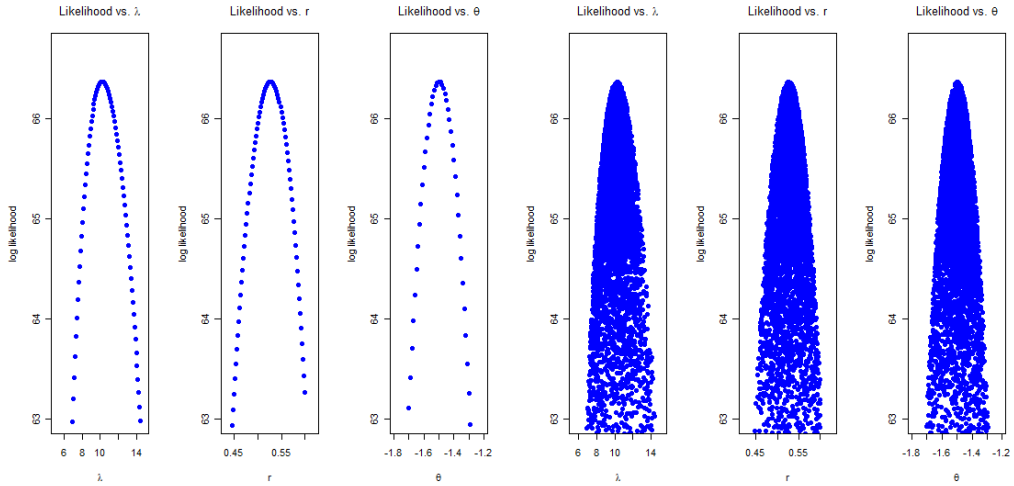

Figure 4.8: Computed Likelihood and MCMC Result for $(k-1) = 2$ No Prior Case

Parameter	MCMC	MLE
$(k-1) = 2$ Case		
λ	10.2 ± 2.9	10.2
r	0.528 ± 0.060	0.527
θ	-1.50 ± 0.16	-1.50
$(k-1) = 3$ Case		
λ_1	112 ± 32	111
λ_2	9.4 ± 2.7	9.4
r	0.9345 ± 0.0078	0.9355
s	0.313 ± 0.037	0.312
a	0.760 ± 0.053	0.764
α	0.87 ± 0.32	0.86
β	3.18 ± 0.42	3.17
γ	2.20 ± 0.30	2.21

Table 4.5: MCMC Parameter Estimates for Flat Prior Cases

Λ_Q and Γ_Q are fixed to values based on our prior beliefs. For the informative priors case, we examined the $(k-1) = 3$ case only. Table 4.6 shows the set

values of the parameters for Λ_Q and Γ_Q. These values were chosen near the values in table 4.2 which formed $\mathcal{S}$. Three Q's were constructed with different

Parameter	Set Value
λ_1	115
λ_2	12
r	0.93544
s	0.3123
a	0.764
α	0.876
β	3.186
γ	2.20

Table 4.6: Λ_Q and Γ_Q Parameters

M's where $M = \mathrm{tr}Q$. M is number of complex Bingham points generated to form the Q's. $M_1 = 500$, $M_2 = 50$, and $M_3 = 5$. The Q's were

$$Q_1 = \begin{pmatrix} 146.9 & \text{-57.1-55.3i} & \text{-162.9+15.3i} \\ \text{-57.1+55.3i} & 60.0 & \text{81.0-94.9i} \\ \text{-162.9-15.3i} & \text{81.0+94.9i} & 293.2 \end{pmatrix}$$

$$Q_2 = \begin{pmatrix} 12.01 & \text{-4.91-4.20i} & \text{-13.78+2.10i} \\ \text{-4.91+4.20i} & 6.08 & \text{9.01-9.73i} \\ \text{-13.78-2.10i} & \text{9.01+9.73i} & 31.91 \end{pmatrix}$$

$$Q_3 = \begin{pmatrix} 1.567 & \text{-0.584-0.575i} & \text{-1.670+0.261i} \\ \text{-0.584+0.575i} & 0.642 & \text{0.802-0.946i} \\ \text{-1.670-0.261i} & \text{0.802+0.946i} & 2.791 \end{pmatrix}$$

Figures 4.10, 4.11, and 4.12 compare the computed likelihood contours and MCMC results. The appendix contains the trace plots and diagnostics. Table 4.7 shows the estimates of the parameters.

As in all the other cases, we concluded that the MCMC was properly sampling the joint posterior. However, the influence of the priors was

Parameter	Q_1	Q_2	Q_3
λ_1	37.6 ± 3.3	16.5 ± 3.3	35.7 ± 9.6
λ_2	11.8 ± 1.0	9.2 ± 1.9	9.6 ± 2.6
r	0.9436 ± 0.0038	0.923 ± 0.043	0.941 ± 0.014
s	0.279 ± 0.024	0.213 ± 0.087	0.287 ± 0.068
a	0.737 ± 0.020	0.568 ± 0.090	0.738 ± 0.067
α	-0.52 ± 0.20	0.79 ± 0.49	0.60 ± 0.65
β	3.45 ± 0.28	-1.74 ± 0.72	2.96 ± 0.85
γ	-2.51 ± 0.17	2.63 ± 0.35	2.46 ± 0.55

Table 4.7: MCMC Parameter Estimates for Informative Priors

somewhat surprising. In all three cases, λ_1 was substantially reduced. The argument parameters α, β, and γ also showed a good amount of variation. The modulus parameters r, s, and a showed more relative stability. λ_2 seemed to be the most stable parameter.

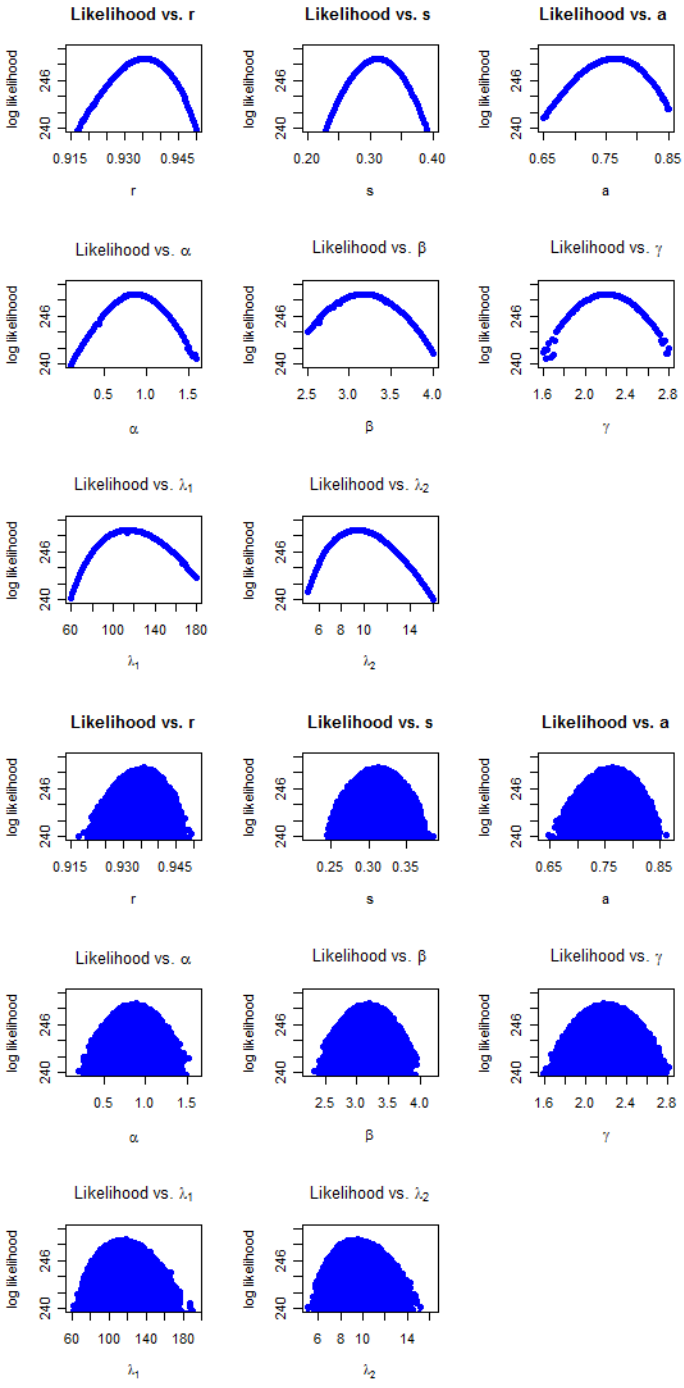

Figure 4.9: Computed Likelihood and MCMC Result for $(k-1) = 3$ No Prior Case

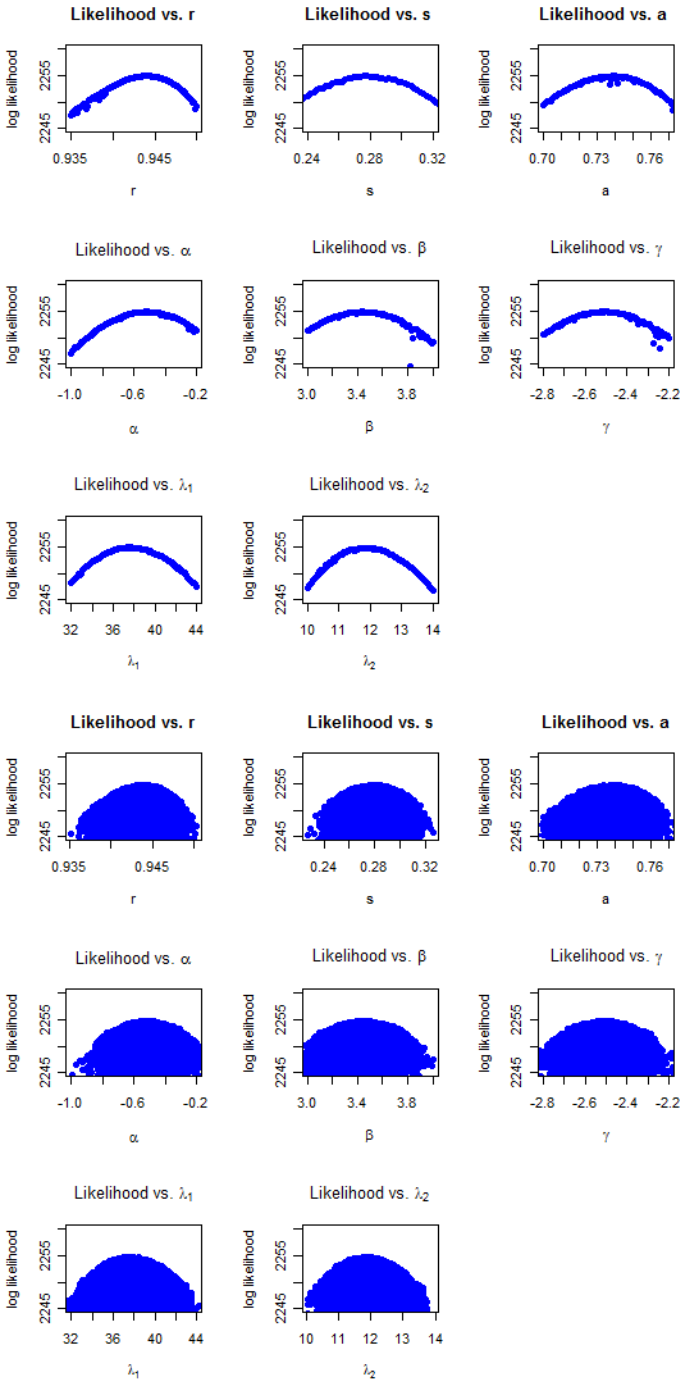

Figure 4.10: Computed Posterior and MCMC Result for Q_1 Prior Case

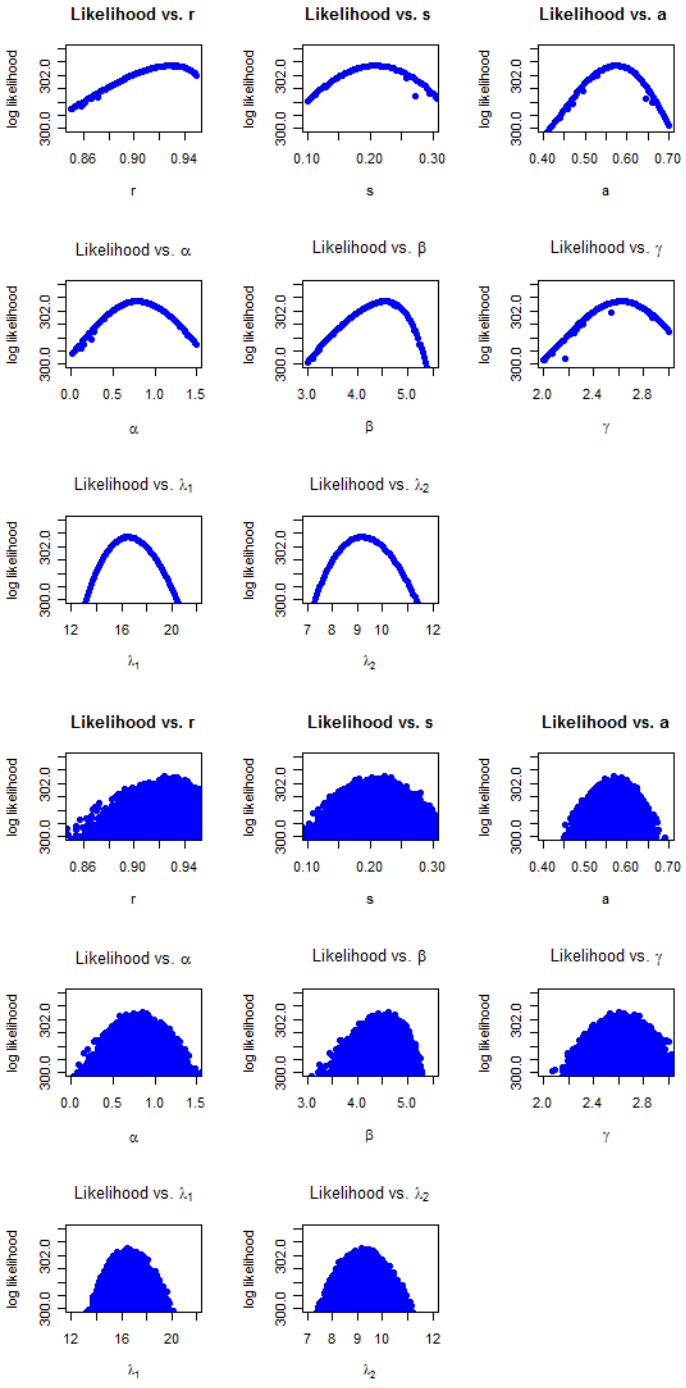

Figure 4.11: Computed Posterior and MCMC Result for Q_2 Prior Case

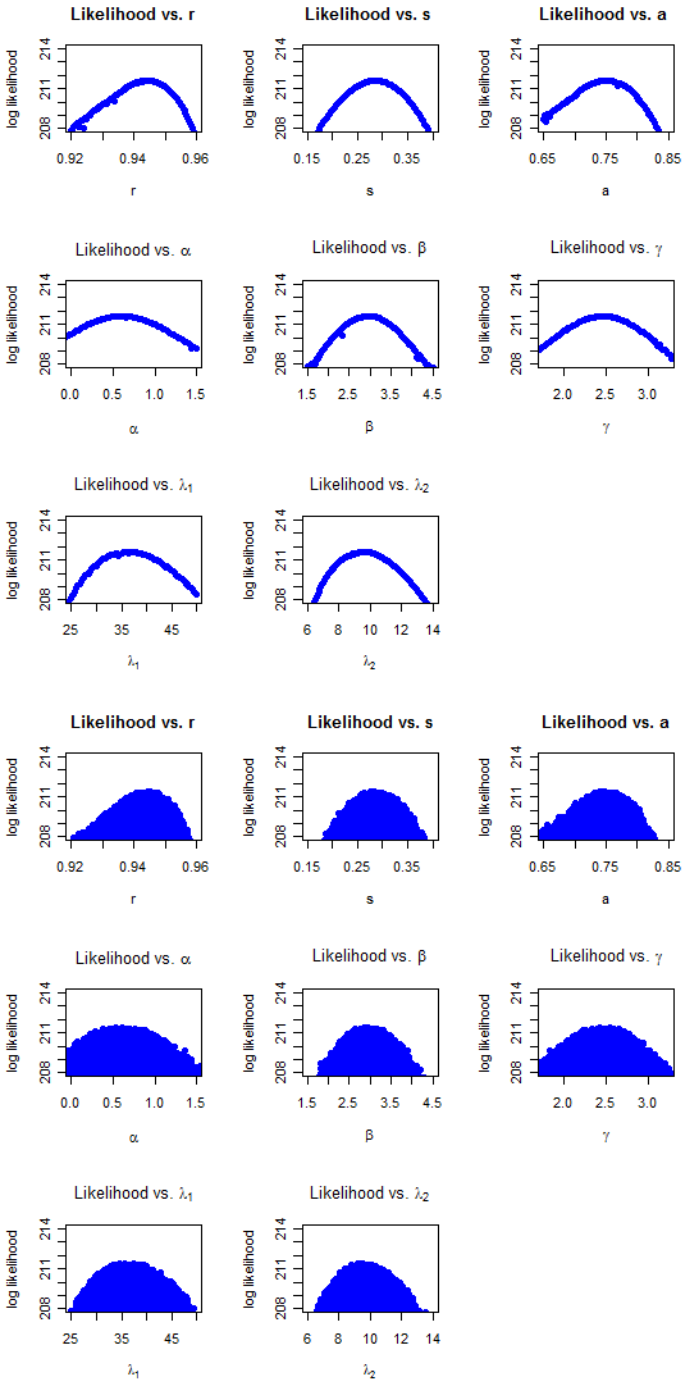

Figure 4.12: Computed Posterior and MCMC Result for Q_3 Prior Case

Chapter 5

Summary

Previous analysis of data arising from a complex Bingham involved classical techniques such as MLE. Bayesian inferences were restricted to the MAP estimator. In this report, a full Bayesian analysis was done on the complex Bingham distribution. We found it convenient to spectral decompose the Hermitian parameter matrix into its eigenvalues and eigenvectors. The joint posterior was examined, using a new conjugate prior. A Gibbs type MCMC was utilized to sample the joint posterior. A Metropolis-Hastings algorithm sampled the conditional posterior of the eigenvalues. Because the eigenvectors are orthogonal to each other, a technique termed "successive conditional Monte Carlo sampler" was developed. The individual components of the Gibbs MCMC were verified on randomly generated data of a known parameter. The Gibbs sampler of the joint posterior was verified in the same manner. Both flat and informative priors were investigated. The algorithm developed in this report is successful in sampling from the posterior distribution when data arise from a complex Bingham distribution.

Appendices

Appendix A

Complex Algebra

A.1 Complex Numbers

We briefly review some properties of complex numbers. Complex numbers can be written in rectangular form as

$$z = A + iB$$

where $A, B \in \mathbb{R}$ and $i^2 = -1$. The real and imaginary parts of z are $\mathrm{Re}(z) = A$ and $\mathrm{Im}(z) = B$, respectively. The complex conjugate is defined as

$$z^* = A - iB.$$

Complex numbers can also be written in polar form as

$$z = re^{i\theta}$$

where r is called the modulus and θ is called the argument. The polar and rectangular forms are connected by Euler's relation

$$e^{i\theta} = \cos(\theta) + i\sin(\theta).$$

The modulus or magnitude is

$$\mathrm{Mod}(z) = |z| = r = \sqrt{A^2 + B^2}$$

while the argument is

$$\mathrm{Arg}(z) = \theta = \tan^{-1}\left(\frac{B}{A}\right).$$

When adding or subtracting, the rectangular form is easier to use.

$$z_1 + z_2 = (A_1 + A_2) + i(B_1 + B_2)$$

When multiplying or dividing, the polar form is easier.

$$z_1 * z_2 = r_1 r_2 e^{i(\theta_1+\theta_2)}$$

A.2 Complex Vectors & Matrices

A complex vector is just a vector with complex components. The magnitude of a complex vector is

$$|\vec{z}| = \sqrt{\sum |z_i|^2}$$

just as in the real case. The only difference is z_i is complex and $|z_i|^2 = r_i^2$. This formulation ensures that the magnitude of a complex vector is positive real. Dot products (or inner products) of complex vectors are

$$\vec{z_1} \cdot \vec{z_2} = \sum_i^m z_{1,i}^* z_{2,i} \tag{A.1}$$

where m is the length of the vector. The result is a complex number.

A vector in matrix notation becomes an mx1 matrix where m is the length of the vector. The dot product of two vectors in matrix notation is

$$\vec{z_1} \cdot \vec{z_2} = z_1^* z_2 = \begin{bmatrix} z_{1,1}^* & z_{1,2}^* & \cdots & z_{1,m}^* \end{bmatrix} \times \begin{bmatrix} z_{2,1} \\ z_{2,2} \\ \vdots \\ z_{2,m} \end{bmatrix}$$

where * denotes the complex conjugate transpose. The complex conjugate transpose of a vector is a 1xm matrix.

Vectors can also be multiplied to form matrices. This operation is called the outer product. For example $z_1 z_2^*$ can be viewed as an mx1 matrix multiplied by a 1xm matrix. This operation results in a mxm matrix. The complex conjugate transpose of a matrix is denoted

$$A^* = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots \\ A_{2,1} & \ddots & \\ \vdots & & \end{bmatrix}^* = \begin{bmatrix} A_{1,1}^* & A_{2,1}^* & \cdots \\ A_{1,2}^* & \ddots & \\ \vdots & & \end{bmatrix}.$$

Note that $z_1 z_1^*$ is equal to its complex conjugate transpose.

$$z_1 z_1^* = \begin{bmatrix} z_{1,1}z_{1,1}^* & z_{1,1}z_{1,2}^* & \cdots & z_{1,1}z_{1,m}^* \\ z_{1,2}z_{1,1}^* & z_{1,2}z_{1,2}^* & & \vdots \\ \vdots & & \ddots & \\ z_{1,m}z_{1,1}^* & \cdots & & \end{bmatrix}$$

For real matrices there are a couple of special forms. A symmetric matrix is one where

$$A_{ij} = A_{ji} \text{ or } A^T = A,$$

and an orthogonal matrix is one where

$$AA^T = A^T A = I$$

where ^{T} is the transpose and I is the identity matrix. The complex analogs to these matrices are the Hermitian and Unitary matrices, respectively. A Hermitian matrix is one where

$$A_{ij} = A_{ji}^* \text{ or } A^* = A,$$

and a Unitary matrix is one where

$$AA^* = A^*A = I.$$

A general NxN complex matrix has $2N^2$ degrees of freedom: N^2 complex numbers. A Hermitian or Unitary matrix has N^2 degrees of freedom. A Hermitian matrix has N real diagonal elements and $\frac{N(N-1)}{2}$ complex elements below the diagonal. The elements above the diagonal are just the complex conjugate of those below. For a Unitary matrix, the easiest way to see the degrees of freedom is to start with $2N^2$ degrees of freedom of a general matrix. There are N^2 constraints since the columns are orthonormal to each other. Therefore, a Unitary matrix has $2N^2 - N^2 = N^2$ degrees of freedom.

The trace of matrix A is

$$\mathrm{tr}A = \sum A_{ii}.$$

When taking the trace of two matrices, the order does matter

$$\mathrm{tr}(ABC) \neq \mathrm{tr}(BAC)$$

which should be clear since matrix multiplication does not commute. However, the trace operation is invariant under cyclic permutations.

$$\mathrm{tr}(ABC) = \mathrm{tr}(BCA) = \mathrm{tr}(CAB)$$

As long as the order is maintained, the trace will be the same.

In linear algebra, the spectral decomposition (or eigendecomposition) of a matrix is a method to write matrices in terms of their eigenvalues and

eigenvectors. Only diagonalizable matrices can be decomposed this way. For an NxN complex matrix A,

$$A = U\Sigma U^{-1}.$$

Σ is a diagonal NxN matrix containing the eigenvalues of A, $\Sigma_{ii} = \lambda_i$. U is an NxN matrix whose columns are the eigenvectors. In the case of a real symmetric matrix, B, the spectral decomposition is

$$B = U\Sigma U^T$$

where U is an orthonormal matrix. For a Hermitian matrix, the spectral decomposition is

$$H = \Gamma\Lambda\Gamma^*$$

where Γ is a Unitary matrix. In terms of the individual eigenvectors and eigenvalues,

$$H = \sum \lambda_i \gamma_i \gamma_i^*.$$

Appendix B

Numerical Algorithms

B.1 Optimization

Numerical calculations of the MLE's involve non-linear optimization schemes. For an overview of these methods, we refer to chapter 10 in [43]. The `optim` function in `R` is an all-purpose optimization routine containing several algorithms. We utilized two of these algorithms: L-BFGS-B[9] and Nelder-Mead[41]. The L-BFGS-B is a bounded version of the limited memory Broyden, Fletcher, Goldfarb, and Shanno (L-BFGS) algorithm[8][18][22][46]. L-BFGS-B is a quasi-Newtonian method using function and gradient values. This algorithm was used in the $(k-1) = 2$ case where bounds on the parameters were easy to define. The Nelder-Mead method is a simplex method using only function values. Nelder-Mead is more robust than L-BFGS-B but is slower to converge. This method was used in the $(k-1) = 3$ case. The bounds on the modulus parameters r and s were difficult to assign.

B.2 Calculation of Null Space Basis

To calculate the null space basis we used the singular value decomposition (SVD) approach. We refer to chapter 2 in [43] for a more thorough

explanation. SVD is a more general form of the spectral decomposition. It can be used on rectangular as well as square matrices. The SVD of an MxN matrix A is

$$A = U\Sigma V^*$$

where U is an MxM unitary matrix, Σ is an MxN diagonal matrix, and V is an NxN unitary matrix. Assume that $n > m$ and $n - m = c$. The null space basis would be the last c columns of V. This remark assumes that all the diagonal elements of Σ are not zero. If some of them are, then the corresponding columns of V would be including in the null space basis.

B.3 Generating Random Unitary Matrices

We denote the group of NxN Unitary matrices as $U(N)$. A uniform distribution of random unitary matrices is when each matrix in $U(N)$ has equal probability. This definition is sound as long as the group is compact, which $U(N)$ is. Otherwise, the probability density function would not integrate to 1. The algorithm below was introduced in [38] and was shown to sample Unitary matrices correctly.

1. Construct a NxN matrix with each element sampled from a complex standard normal distribution

2. QR decompose this matrix

3. Create diagonal matrix Λ with elements $\Lambda_{ii} = R_{ii}/|R_{ii}|$

4. Matrix $Q * \Lambda$ will be from $U(N)$, uniformly.

Appendix C

R-Code

All computational calculations were done in `R`. The major components of our code are publicly available at

`http://webspace.utexas.edu/leurh/www/cbingham/`

. The following routines are available.

- MH.R: Metropolis-Hastings sampler of the eigenvalues and calculator of the log-likelihood.
- SCMC.R: Successive conditional Monte Carlo sampler of the eigenvectors
- MCMC.R: Gibbs sampler of the joint posterior utilizing MH.R and SCMC.R.

MH.R uses the `msm`[24] library to sample from a truncated normal. The convergence diagnostics and plotting of MCMC's was done with the `coda`[42] package. The `coda` package includes routines to calculate the means, standard deviations, and quantile estimates. It also includes plot routines to produce the trace plots and density estimates. Several convergence tests are part of

the package including. the Geweke test (`geweke.diag`). Calculations of the MLE's were done with the `optim` routine in the `stats` package.

Appendix D

MCMC Diagnostics

This section contains the trace plots and density estimates for all cases. Each trace plot is shown with any burn-in elements already removed. This situation was only relevant in the informative prior case where the initial guesses were far from the posterior peak. A table with the Geweke convergence test scores are shown. All chains showed scores with magnitude below 2.

Figures D.1 and D.2 are the trace plots and marginal density estimates from M-H testing. Each chain demonstrates good mixing and no burn-in effects. The Geweke convergence test z-scores were 0.97 and $\{1.86, 1.33\}$ for the $(k-1) = 2$ and $(k-1) = 3$ cases, respectively. Figures D.3 and D.4 are the trace plots and marginal density estimates for SCMC testing. Each chain exhibits good mixing and no burn-in effects. Table D.1 shows the Geweke convergence test z-scores. Figures D.5 and D.6 are the trace plots and marginal density estimates for MCMC testing in the flat prior case. Figures D.7, D.8, and D.9 show these plots for the informative prior cases. Tables D.2 nad D.3 shows the Geweke convergence test z-scores.

Parameter	Z-Score
$(k-1) = 2$ Case	
r	-1.7
θ	-0.14
$(k-1) = 3$ Case	
r	0.014
s	0.45
a	-0.55
α	0.15
β	-0.068
γ	0.20

Table D.1: Geweke Scores for SCMC

Parameter	Z-Score
$(k-1) = 2$ Case	
λ	-0.62
r	1.2
θ	-1.6
$(k-1) = 3$ Case	
λ_1	0.57
λ_2	1.53
r	-1.1
s	1.0
a	-0.54
α	-0.16
β	-0.13
γ	-0.18

Table D.2: Geweke Scores for Flat Prior Case

Parameter	Z-scores		
	Q_1	Q_2	Q_3
λ_1	0.84	0.97	-0.013
λ_2	-0.22	0.88	-0.65
r	-1.0	0.20	0.72
s	1.2	-0.83	-0.012
a	0.65	0.35	-0.024
α	0.51	-1.2	0.90
β	0.29	-0.90	0.71
γ	-0.22	1.3	-1.0

Table D.3: Geweke Scores for Informative Prior Cases

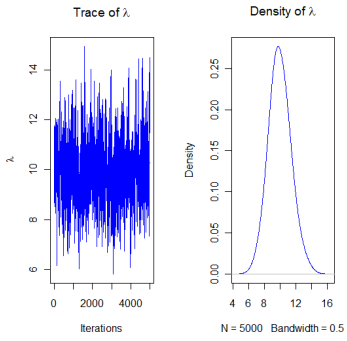

Figure D.1: Trace and Density Plots for λ

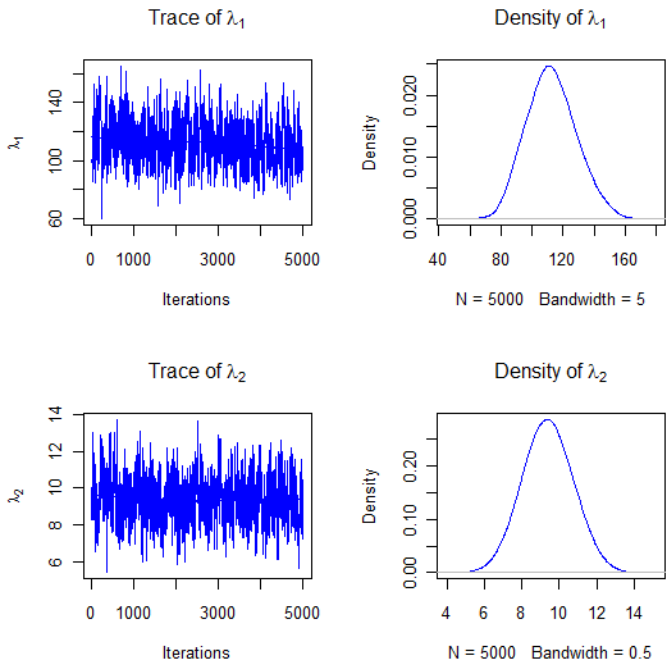

Figure D.2: Trace and Density Plots for Λ

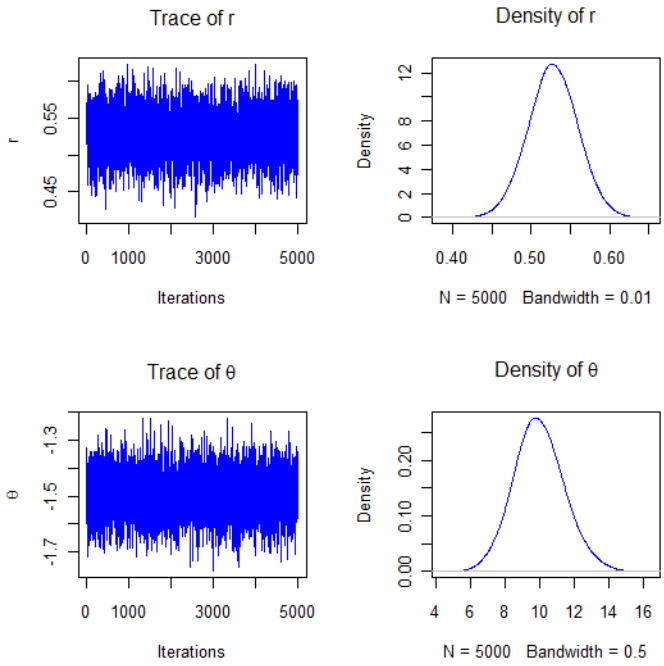

Figure D.3: SCMC Trace and Density Plots for r and θ

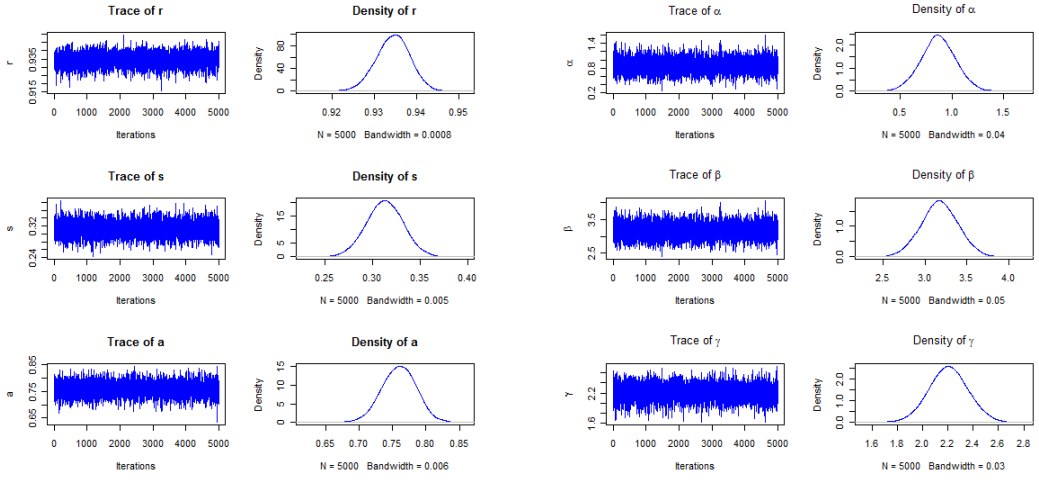

Figure D.4: SCMC Trace and Density Plots for $(k-1) = 3$ case

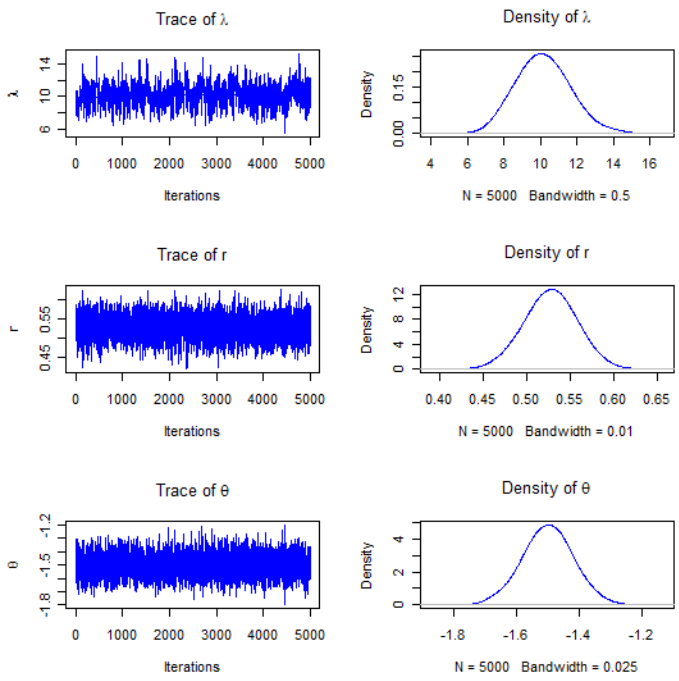

Figure D.5: MCMC Trace and Density Plots for $(k-1) = 2$ Flat Priors Case

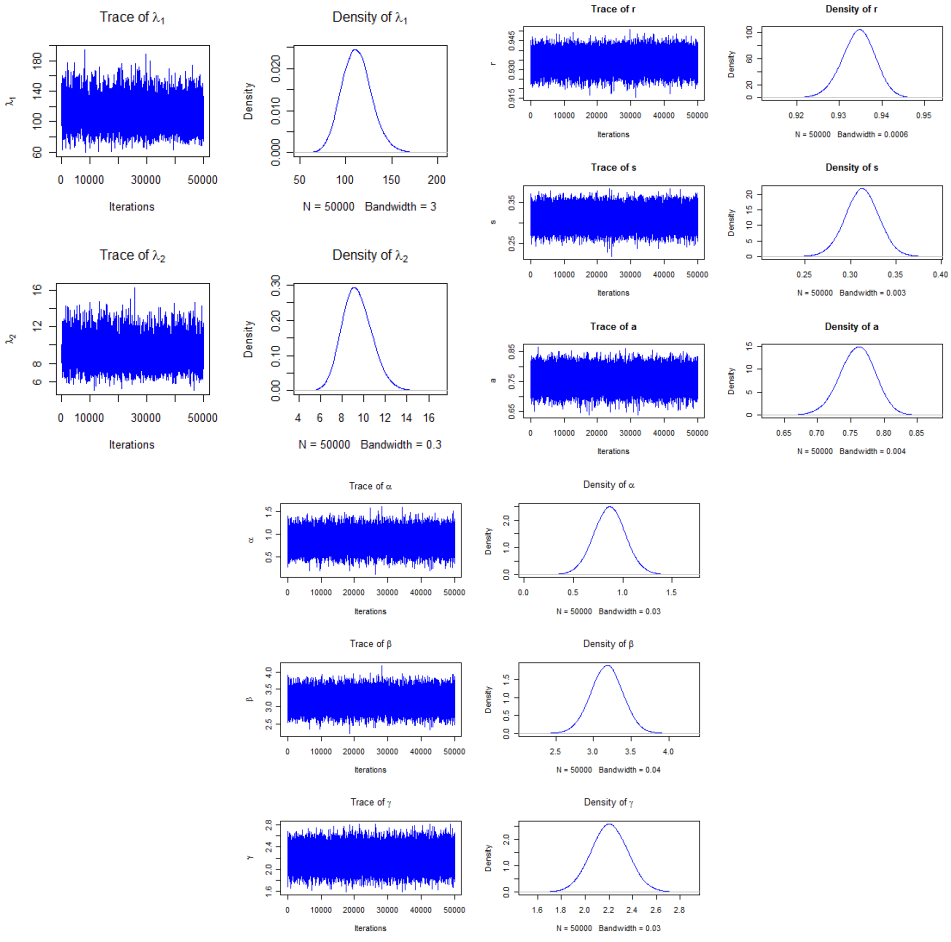

Figure D.6: MCMC Trace and Density Plots for $(k-1) = 3$ Flat Priors Case

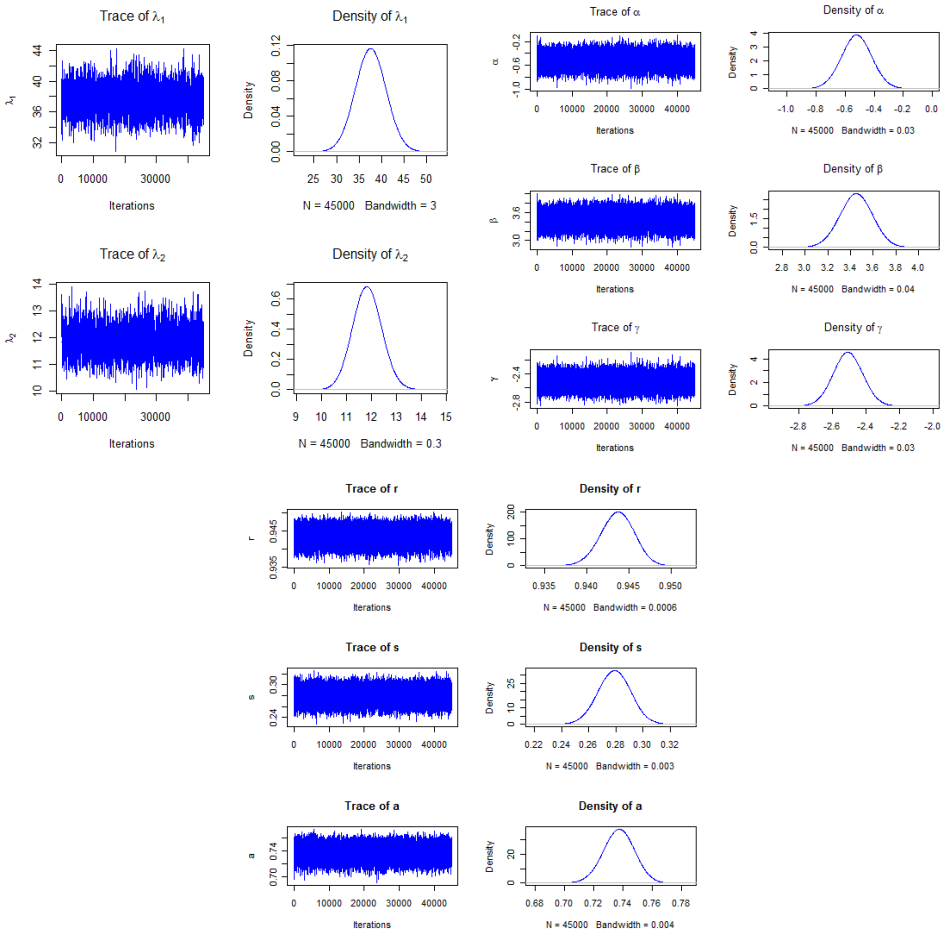

Figure D.7: Trace and Density Plots for Q_1 Case

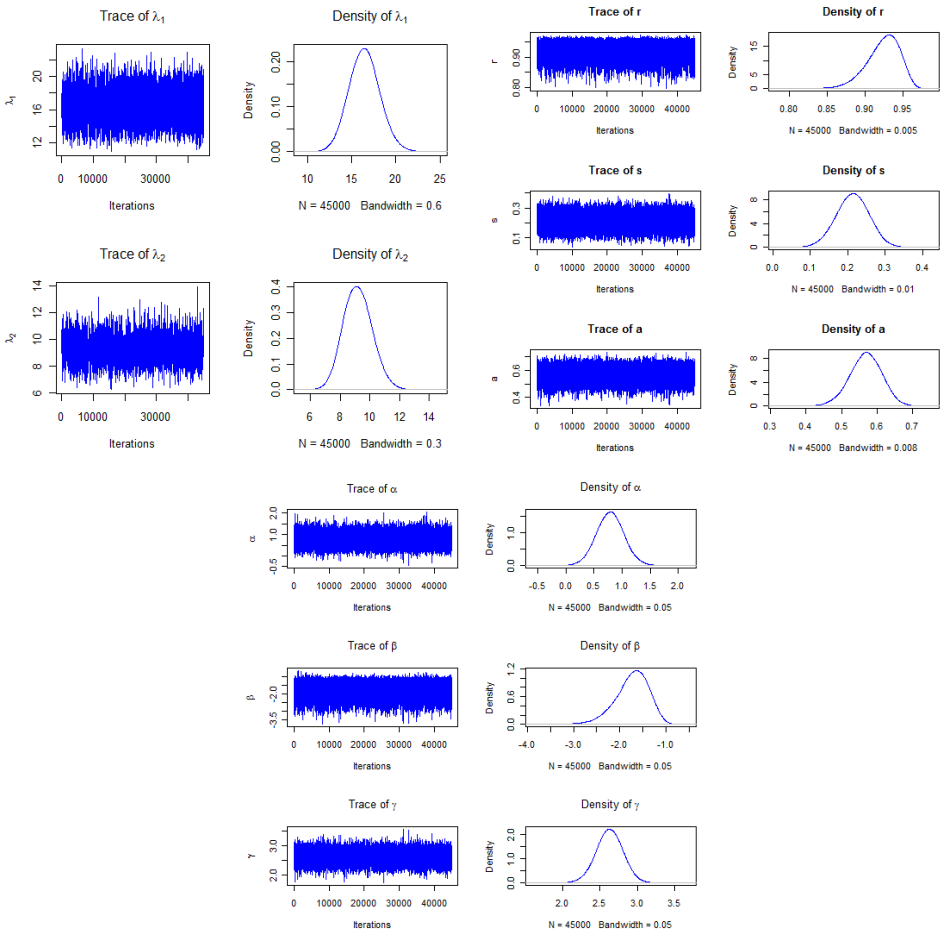

Figure D.8: Trace and Density Plots for Q_2 Case

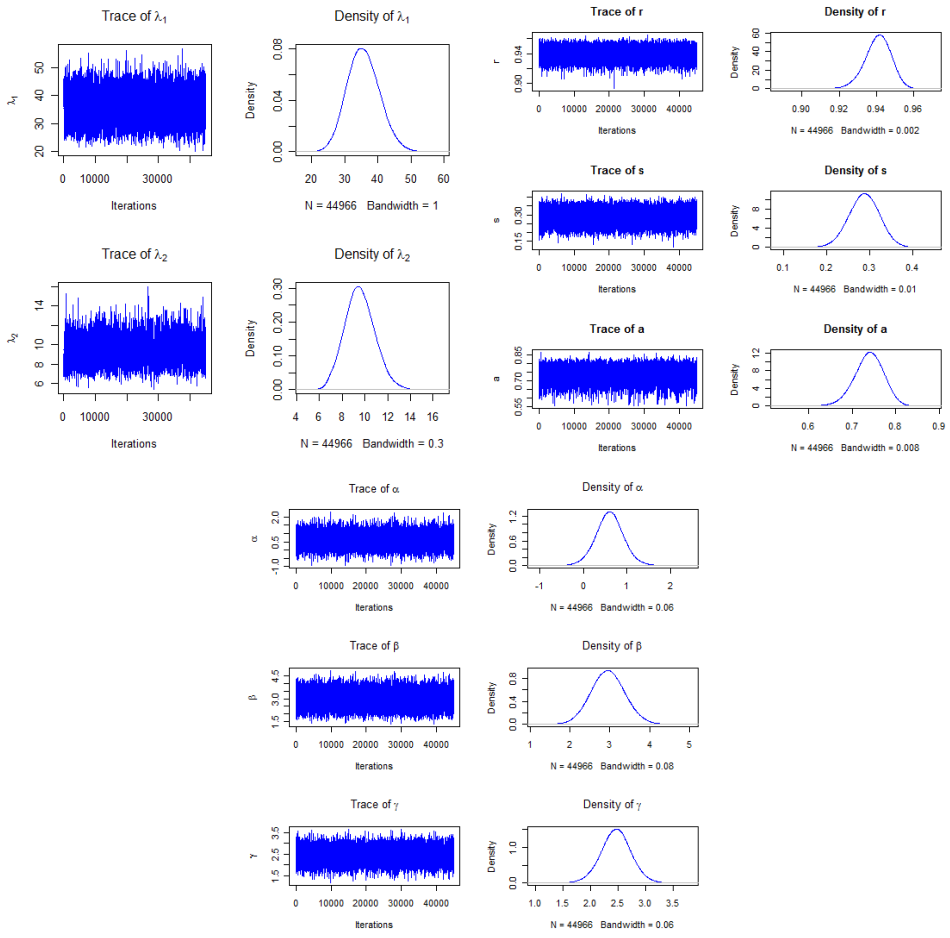

Figure D.9: Trace and Density Plots for Q_3 Case

Bibliography

[1] Getulio J. A. Amaral and Andrew T. A. Wood. Empirical likelihood methods for two-dimensional shape analysis. *Biometrika*, 97(3):757–764, 2010.

[2] C. R. Anderson. *Object recognition using statistical shape analysis*. PhD thesis, University of Leeds, 1997.

[3] J. Anemüllera, T. J. Sejnowski, and S. Makeig. Complex independent component analysis of frequency-domain electroencephalographic data. *Neural Networks*, 16:1311–1323, 2003.

[4] F. L. Bookstein. Size and shape spaces for landmark data in two dimensions (with discussion). *Statistical Science*, 1:181–242, 1986.

[5] F. L. Bookstein. Biometrics, biomathematics, and the morphometric synthesis. *Bulletin of Mathematical Biology*, 58:313–365, 1996.

[6] F.L. Bookstein. *Morphometric tools for landmark data*. Cambridge University Press, 1991.

[7] S. Broadbent. Simulating the ley hunter. *Journal of the Royal Statistical Society, Series A*, 143:109–140, 1990.

[8] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 1970.

[9] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Scientific Computing*, 16:1190–1208, 1995.

[10] V. Calhoun, T. Adali, and L. Yiou. Independent component analysis of complex-valued functional magnetic resonance imaging data by complex nonlinearities. In *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, pages 984 –987 Vol. 1, april 2004.

[11] Cheng-Ying Chou and M.A. Anastasio. Statistical properties of x-ray phase-contrast tomography. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 6648–6650, sept. 2009.

[12] M. K. Cowles and B. P. Carlin. The diffusion of shape. *Journal of the American Statistical Association*, 91(434):883–904, 1996.

[13] I. L. Dryden. *Statistical Shape Analysis*. John Wiley & Sons, Inc., 1998.

[14] I. L. Dryden and K. V. Mardia. Multivariate shape analysis. *Sankhya Series A*, 55:460–480, 1993.

[15] Ian L. Dryden. Statistical analysis on high-dimensional spheres and shape spaces. *The Annals of Statistics*, 33(4):1643 – 1665, 2005.

[16] Ian L. Dryden, Alfred Kume, Huiling Le, and Andrew T. A. Wood. A multi-dimensional scaling approach to shape analysis. *Biometrika*, 95(4):779–798, 2008.

[17] J. Eriksson and V. Koivunen. Complex-valued ica using second order statistics. In *Machine Learning for Signal Processing, 2004. Proceedings of the 2004 14th IEEE Signal Processing Society Workshop*, pages 183–192, 29 2004-oct. 1 2004.

[18] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.

[19] J. Geweke. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. *Bayesian Statistics*, 4:169–193, 1992.

[20] Geof H. Givens and Jennifer A. Hoeting. *Computational Statistics*. John Wiley & Sons, Inc., 2005.

[21] S. L. Goh, M. Chen, D.H. Popovic, K. Aihara, D. Obradovic, and D. P. Mandic. Complex-valued forecasting of wind profile. *Renewable Energy*, 31:1733–1750, 2006.

[22] D. Goldfarb. A family of variable metric updates derived by variational means. *Mathematics of Computation*, 24:23–26, 1970.

[23] C. R. Goodall. Procrustes methods in the statistical analysis of shapes. *Journal of the Royal Statistical Society, Series B*, 53:285–339, 1991.

[24] Christopher H. Jackson. Multi-state models for panel data: The msm package for R. *Journal of Statistical Software*, 38(8):1–29, 2011.

[25] D. G. Kendall. The diffusion of shape. *Advances in Applied Probability*, 9:428–430, 1977.

[26] D. G. Kendall. Shape manifolds, procrustean metrics and complex projective spaces. *Bulletin of the London Mathematical Society*, 16:81–121, 1984.

[27] D.F. Kendall and W. S. Kendall. Alignments in two dimensional random sets of points. *Advances in Applied Probability*, 12:380–424, 1980.

[28] J. T. Kent. The complex bingham distribution and shape analysis. *Journal of the Royal Statistical Society, Series B*, 56:285–299, 1994.

[29] J. T. Kent. Data analysis for shapes and images. *Journal of Statistical Inference and Planning*, 57:181–193, 1997.

[30] J. T. Kent, P. Constable, and F. Er. Simulation for the complex bingham distribution. *Statistics and Computing*, 14(1):53–57, 2004.

[31] C. G. Khatri and K. V. Mardia. The von mises-fisher distribution in orientation statistics. *Journal of the Royal Statistical Society, Series B*, 39:95–106, 1977.

[32] A. Kume and Andrew T.A. Wood. On the derivatives of the normalising constant of the bingham distribution. *Statistics & Probability Letters*, 77(8):832 – 837, 2007.

[33] D. P. Mandic, S. Javidi, S. L. Goh, A. Kuh, and K. Aihara. Complex-valued prediction of wind profile using augmented complex statistics. *Renewable Energy*, 34:196–201, 2009.

[34] D.P. Mandic, S. Javidi, G. Souretis, and V.S.L. Goh. Why a complex valued solution for a real domain problem. In *Machine Learning for Signal Processing, 2007 IEEE Workshop on*, pages 384 –389, aug. 2007.

[35] K. V. Mardia and I. L. Dryden. The statistical analysis of shape data. *Biometrika*, 76:71–282, 1989b.

[36] K. V. Mardia and I. L. Dryden. The complex watson distribution and shape analysis. *Journal of the Royal Statistical Society, Series B*, 61:913–926, 1999.

[37] Eduardo Martínez-Montes, Elena R. Cuspineda-Bravo, Wael El-Deredy, José M. Snchez-Bornot, Agustn Lage-Castellanos, and Pedro A. Valdès-Sosa. Exploring event-related brain dynamics with tests on complex valued time-frequency representations. *Statistics in Medicine*, 27:2922–2947, 2008.

[38] Francesco Mezzadri. How to generate random matrices from the classical compact groups. *Notices of the AMS*, 54:592–604, 2007.

[39] A. C. Micheas and D. K. Dey. Assessing shape differences in populations of shapes using the complex watson shape distribution. *Journal of Applied Statistics*, 32(2):105 – 116, 2005.

[40] F. Neeser and J. Massey. Proper complex random processes with applications to information theory. *IEEE Transactions on Information Theory*, 39:1293–1302, 1992.

[41] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[42] Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, March 2006.

[43] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.

[44] P. Schreier and L. Scharf. Second-order analysis of improper complex randomvectors and processes. *IEEE Transactions on Signal Processing*, 51:714–725, 2003.

[45] Peter J. Schreier and Louis L. Scharf. *Statistical Signal Processing of Complex-Valued Data*. Cambridge University Press, 2010.

[46] David F. Shanno. Conditioning of quasi-newton methods for function minimization. *Math. Comput.*, 24:647–656, 1970.

[47] R. Southworth, K. V. Mardia, and C. C. Taylor. Transformation- and label-invariant neural network for the classification of landmark data. *Journal of Applied Statistics*, 27(2):205 – 215, 2000.

[48] A. Tarighat and A.H. Sayed. Least mean-phase adaptive filters with application to communications systems. *IEEE Signal Processing Letters*, 11:220–223, 2004.

[49] B. Widrow, J. McCool, and M. Ball. The complex lms algorithm. *Proceedings of the IEEE*, 63(4):719 –720, april 1975.

[50] Siwei Yang, Sandra Gtze, Julio Mateos-Langerak, Roel van Driel, Roland Eils, and Karl Rohr. Analyzing the variability of the 3d structure of chromatin fiber using statistical shape theory. In Fred Hamprecht, Christoph Schnrr, and Bernd Jhne, editors, *Pattern Recognition*, volume 4713 of *Lecture Notes in Computer Science*, pages 497–506. Springer Berlin / Heidelberg, 2007.